**3740 Calc Manual**

**Calling The Interpreter**

The user can call the interpreter by typing the following into Dr. Racket:

(uofl)

This command opens up a prompt displaying “UofL>” and will read the entire line that the user enters.

The interpreter then parses it into tokens and lexers, which is used to do further calculations. This is automatically handled by UofL.

**Declaring Variables**

Variables must be defined before its usage. A name and datatype must be initialized at the same time as follows:

#definevari (variable name) (variable datatype)

For example:

* #definevari intput1 integer :
  + This statement is used to define a variable with a name as input1 as datatype as integer.
* #definevari input2 float:
  + This statement is used to define a variable with a name as input2 as datatype as float.
* #definevari input2 boolean:
  + This statement is used to define a variable with a name as input3 as datatype as boolean.

**Declaring Functions:**

When defining a function you must use predefined variables when declaring a function as follows:

#definefunc (function name) (predefined variable …. predefined variable) #definefunc

For Example:

* #definefunc main x y z

x = y\* z

output x

#definefunc

**Calling Functions:**

To call a function, it must be previously defined. The function name is then to be typed into Dr. Racket. Calling the function must be done in the following form:

(function name)

**Arithmetic Expressions**

Arithmetic expressions can be typed into Dr. Racket with a traditional mathematical form. The expression will be evaluated following tradition precedence rules. The following displays the form:

(predefined variable ) = (predefined variable) ( op-token) (predefined variable)

| (pre-defined variable) (op-token)

| (op-token) (pre-defined variable)

Op-token:

* +, -, /, \*, ^, NEG, !, =

For example:

* a = (b + 2) \* 5 + 2^b or b = b + 4 \* 6 / c + 7 \* c

**Relational expression:**

Relational expressions will return a true or false value (boolean). Relational expressions are called with the following form:

(predefined variable ) = (predefined variable) ( op-token) (predefined variable)

Op-token:

* ==, >=, <=, <>, >, <

For example:

* a== b or a >=b or a <= b or a<> b or a<b or a>b

**Selection statement:**

Variables must be predefined before its use. Selection statements are called with the following form:

if (relational expression ) then (arithmetic expression) elseif (relational expression) then (arithmetic expression) endif

For example: if (a <6) then b = 8 elseif (a > 6) then b = 10 endif

**Iterative statement:**

Iterative statements are called with the following form:

for (assignment expression) to ( integer) stepsize ( integer) do (body containing statements or arithmetic operations or functions ) endfor

For example:

For I = 1 to stepsize 2 do

a = a\*2

endfor

**Input statement:**

To input a value for a variable, the following is to be typed into Dr. Racket:

input (pre-defined variable)

The interpreter throws a command :

Input a value for a: (user input a value)

For example:

input a

This will prompt:

“Input a value for a:”

The user can then enter a value:

10

**Output statement:**

To output a value for a variable, the following is to be typed into Dr. Racket:

output (pre-defined variable)

For example: output a

The interpreter outputs the value of the variable.

**Reserved Words**

The following are special commands for uofl: The following must be typed into UofL:

* #clear - it clears all the data stored in the interpreter
* #exit - it makes the user come of the (uofl) function
* I, J- used for for loop and it cannot be used as a variable

**Testing Methods**

* If the if else loop isn’t correct then the interpreter gives an error message: Token ‘EOF’ has an error. It’s value: #f
* Unknown parser error: if there is a error but it isn’t the token error
* Variable not defined: if user try to use an undefined variable then interpreter displays that variable is not defined.
* Function not defined: if user try to use an undefined function then interpreter displays that variable is not defined.
* Type error for variable: if user try to compare two different data types then the interpreter outputs this error.

For example:

* if( a== 10 ) then a = 20 // Token ‘EOF’ has an error. It’s value: #f
* Output b // Variable not defined , that is it is used with #definevari or input
* #definevari a integer

#defienvari b bool

a= a +b // Type error for variable